

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/305407573>

Path-following NMPC for Serial-link Robot Manipulators Using a Path-parametric System Reformulation

Conference Paper · June 2016

CITATIONS

0

READS

47

5 authors, including:



Niels van Duijkeren

University of Leuven

3 PUBLICATIONS 2 CITATIONS

SEE PROFILE



Goele Pipeleers

University of Leuven

79 PUBLICATIONS 436 CITATIONS

SEE PROFILE



Moritz Diehl

University of Freiburg

264 PUBLICATIONS 4,604 CITATIONS

SEE PROFILE



Jan Swevers

University of Leuven

51 PUBLICATIONS 226 CITATIONS

SEE PROFILE

Path-following NMPC for Serial-link Robot Manipulators Using a Path-parametric System Reformulation*

Niels van Duijkeren¹, Robin Verschueren², Goele Pipeleers¹, Moritz Diehl², Jan Swevers¹

Abstract—This paper discusses path-following control for robotics, moving a manipulator along a path in Cartesian space, making a trade-off between tracking accuracy and the speed at which the path is followed. We present and validate a nonlinear model predictive control (NMPC) approach suitable for this nonlinear control task. This approach entails a method to model the position of the robot end-effector with respect to the path and, in addition, a reformulation of the robot prediction model in terms of an independent path parameter instead of time. This way, we obtain a convenient parameterization of path properties in the optimal control formulation and many geometric constraints, such as tracking tolerance, transform into simple linear or vector-norm constraints. Numerical simulations illustrate the benefits of this novel NMPC approach in an implementation that employs a direct multiple shooting discretization strategy and the real-time iteration scheme for fast computation of the control law. We show results of closed-loop simulations for a 6-DOF industrial robot executing a writing task, with computation times close to enabling real-time implementation.

I. INTRODUCTION

In robotics, it is often the case that the essence of a given task is described by the path the end-effector should follow. Examples include the control of welding, gluing, assembly robots and CNC machines. Offline methods exist to generate a time-/energy-optimal trajectory aiming at tracking a given geometric path *exactly* [1], [2] and recently an efficient convex reformulation of this problem has been derived [3], [4]. These path tracking methods generate a time-based reference signal that can readily be fed to joint angle controllers of the robotic system. Path following control concerns feedback laws that drive systems to follow a geometric path. In practice this means that the *timing law* (i.e., where to be when on the path) is introduced as control freedom. This is an attractive approach compared to tracking time-based reference signals and has proven to be an appropriate strategy for the control of certain types of nonlinear systems exhibiting unstable zero dynamics [5], [6].

*Niels van Duijkeren en Robin Verschueren are fellows of the TEMPO Initial Training Network. The work leading to these results has been carried out within the framework of FP7-ITN-TEMPO (607 957). This work also benefits from project G0C4515N of the Research Foundation - Flanders (FWO - Flanders), KU Leuven-BOF PFV/10/002 Centre of Excellence: Optimization in Engineering (OPTEC) and the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO).

¹Division of Production engineering, Machine design and Automation, Department of Mechanical Engineering, KU Leuven, Belgium
Niels.vanDuijkeren@kuleuven.be

²Systems Control and Optimization Laboratory, Department of Microsystems Engineering, University of Freiburg, Germany
robin.verschueren@imtek.uni-freiburg.de

For certain tasks, deviations from the desired path within given tolerance are allowed, and this additional freedom can be used to further optimize the robot motion. For example, certain tasks have an intrinsic machining tolerance, and limited deviations from the nominal path might be perfectly acceptable. A controller can be given the freedom to trade off path following accuracy and path following velocity by optimizing the desired position along the path and the distance vector of this point to the robot end-effector. Optimization-based predictive approaches to path following control are of active interest to obtain a feedback law, model predictive path-following control (MPFC) refers to the class of controllers that implement this paradigm [7]. They consider the desired path following velocity (or one of its time derivatives) as an additional control input and thereby allow trading off path following accuracy against the speed with which the following task is completed. Stability results for MPFC have been published, cf. [8], [9], [7]. Similar to approaches in NMPC, stabilizing feedback laws are obtained by choosing an appropriate terminal-state penalty and constraint [8] or, alternatively, by introducing a contraction constraint [9]. Recent results in the field of MPFC also include the real-time implementation on a robot arm, requiring fast computation of the control law [10].

We introduce a control strategy related to MPFC, where the geometric nature of path following is more prominently present in the problem formulation. The crucial difference with previous works on MPFC is that we explicitly express the path following velocity as a function of the states in the prediction model. We employ this expression to perform a transformation of variables for the prediction model to evolve with a path parameter, rather than with time. Hence, path properties such as curvature, but also collision avoidance constraints, can be included in the optimization problem effectively. This path-parametric system reformulation has been introduced in [11], [12] for controlling the planar motion of autonomous vehicles. These works also effectively illustrate how the spatial reformulation enables anti-collision constraints to appear linearly in the optimization. Further applications that express these advantages include research on driver assistance functions for highway driving of long heavy vehicle combinations [13], and time-optimal NMPC on small-scale race cars [14]. The key contribution of this paper consists of a geometric representation of the path following problem in the workspace of a 6-DOF robot manipulator and a validation of the control strategy in a simulation environment. This involves a generalization of the time transformation described above to non-planar motions

and, in addition, a fast implementation of the NMPC strategy. Short computation times of the control law are achieved by employing the real-time iteration (RTI) scheme.

The remainder of this paper is structured as follows. First, in Section II the spatial reformulation of the robot dynamics and the modeling of the position of the end-effector with respect to the path is described. Thereafter in Section III the optimal control problem (OCP) under consideration is described and the different components of the problem are discussed. As an illustration, a scenario of a 6-DOF robot writing a text is presented in Section IV using numerical simulations. A comparison is made between different tuning instances of the control objective trading off path following accuracy and the velocity tracking error. The paper concludes with interesting challenges regarding implementation and further extensions of the proposed method in Section V.

Throughout this paper, we will use the notation $(\cdot)' = \frac{d(\cdot)}{ds}$ and $(\dot{\cdot}) = \frac{d(\cdot)}{dt}$.

II. SPATIAL REFORMULATION OF ROBOT DYNAMICS

To illustrate the spatial reformulation, we apply it to writing task for a robot arm. Let us consider a rigid-body n-DOF serial-link robotic manipulator. Recall that the motion equations of this kind of systems can be written in the form [15]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau \quad (1)$$

where $q, \dot{q}, \ddot{q}, \tau \in \mathbb{R}^n$ are the joint angles, joint velocities, joint accelerations and joint actuation torques, respectively. $M(q)$, $C(q, \dot{q})$ respectively denote the mass matrix and a matrix accounting for Coriolis and centrifugal effects, $G(q)$ is a vector of torques due to gravitation and $F(\dot{q})$ models friction in the joints. The friction term is composed of Coulomb friction $F_s \text{sgn}(\dot{q})$ and viscous friction $F_v \dot{q}$,

$$F(\dot{q}) = F_s \text{sgn}(\dot{q}) + F_v \dot{q}. \quad (2)$$

Below, we present our representation of the path and establish the equation that expresses the progress of the robot end effector along the path. We then use this relation to derive the spatial reformulation of the robot dynamics.

A. Path Representation

Let $\gamma(t)$ be a continuous, sufficiently smooth curve in three-dimensional Euclidean space, and let us assume that the velocity vector $\dot{\gamma}(t) \neq 0$. We introduce the arc length $s(t)$ as the distance traveled along the path. The path $\Gamma = \{\gamma(s) \in \mathbb{R}^3 : s \in [0, l]\}$ is parametrized by its arc length

$$s(t) = \int_0^t \|\dot{\gamma}(x)\|_2 dx. \quad (3)$$

Local properties of the curve are characterized by the curvature κ and the torsion σ . At each point s we define an orthonormal basis frame of three vectors \mathcal{T} , \mathcal{N} and \mathcal{B} , referred to as the tangent, normal and binormal unit vectors. These unit vectors are defined by $\mathcal{T}(s) := \gamma'(s)$,

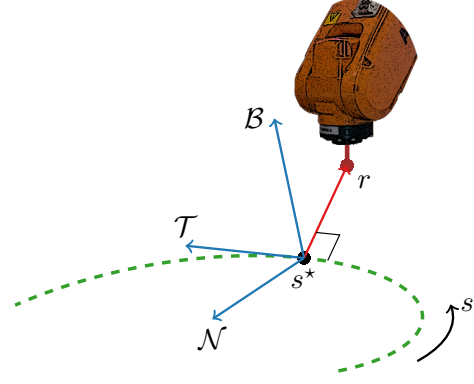


Fig. 1. Illustration of the position of the end-effector with respect to the closest position of the path and the local Frenet-Serret frame in that point.

$\mathcal{N}(s) := \mathcal{T}'(s)/\|\kappa(s)\|_2$ and $\mathcal{B}(s) := \mathcal{T}(s) \times \mathcal{N}(s)$ and satisfy the Frenet-Serret formulas, cf. [16]:

$$\mathcal{T}' = \kappa \mathcal{N}, \quad \mathcal{N}' = -\kappa \mathcal{T} + \sigma \mathcal{B}, \quad \mathcal{B}' = -\sigma \mathcal{N}. \quad (4)$$

Furthermore, let $p(t)$ be the position of the end effector in the inertial world frame, which can be calculated using the forward kinematics of the robot manipulator. Then the point on the path γ closest to $p(t)$ is $\gamma(s^*)$, where

$$r(s, t) = p(t) - \gamma(s) \quad (5)$$

$$s^* = \arg \min_s \frac{1}{2} \|r(s, t)\|_2^2. \quad (6)$$

See Fig. 1 for an illustration of the concept.

As is clear from (5)-(6), finding $s^*(t)$ involves an optimization problem. Since it is undesirable to embed this into a higher-level optimization problem, we determine the temporal evolution of $s^*(t)$ from the optimality conditions for (6). Recall that for unconstrained optimization, the necessary first order optimality condition for (6) is:

$$0 = \frac{d}{ds} \left(\frac{1}{2} \|r(s, t)\|_2^2 \right) \quad (7a)$$

$$= r(s, t)^\top \gamma'(s). \quad (7b)$$

Assuming that the position s^* is known at the initial time-point, we can enforce the solution to be optimal in time by setting the time derivative of the necessary first order optimality condition (7) to zero, i.e.,

$$0 = \frac{d}{dt} \left(r(s, t)^\top \gamma'(s) \right) \quad (8a)$$

$$= (v(t) - \gamma'(s) \dot{s}(t))^\top \gamma'(s) + r(s, t)^\top \gamma''(s) \dot{s}(t), \quad (8b)$$

where $v(t) = \frac{dp(t)}{dt}$. This ultimately gives us a closed formula for the velocity of the point on the path closest to $p(t)$,

$$\dot{s}(t) = \frac{v(t)^\top \mathcal{T}(s)}{1 - \kappa(s) r(s, t)^\top \mathcal{N}(s)}. \quad (9)$$

B. Spatial reformulation

We augment the state vector with $r = [r_x, r_y, r_z]$. Additionally, the orientation ϕ of the end-effector (in *ZYX-Euler angles* [15]) and the joint torque τ are included in the state vector; the latter allows $\dot{\tau}$ to be chosen as control variable. The state vector then reads $\xi = [q^\top, \dot{q}^\top, r^\top, \phi^\top, \tau^\top]^\top \in \mathbb{R}^m$. Using the established representation for the dynamics of the position of the end-effector $p(t)$ with respect to the path, we perform a spatial transformation of the equations of motion:

$$\xi' := \frac{d\xi}{ds} = \frac{d\xi}{dt} \frac{dt}{ds}, \quad (10)$$

with the state vector ξ . For $\dot{s}(t) \neq 0$, we have that $\frac{dt}{ds} = \frac{1}{\dot{s}(t)}$, and therefore

$$\xi' = \frac{1}{\dot{s}(t)} \dot{\xi}. \quad (11)$$

The resulting equations of motion are:

$$f_1(\xi, \dot{\tau}, \ddot{q}_a, s) = \frac{d\xi}{ds} = \frac{\left[\dot{q}^\top, \ddot{q}_a^\top, \dot{p}^\top - \mathcal{T}(s)^\top \dot{s}, \dot{\phi}^\top, \dot{\tau}^\top \right]^\top}{\dot{s}} \quad (12a)$$

$$f_2(\xi, \ddot{q}_a) = 0 = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) - \tau \quad (12b)$$

where the translational velocity of the end-effector can be written as $\dot{p} = J_p(q)\dot{q}$ and the rotational velocity as $\dot{\phi} = J_\phi(q, \phi)\dot{q}$; with $J_a(q, \phi) = [J_p(q)^\top, J_\phi(q, \phi)^\top]^\top$ the analytical robot Jacobian, and \dot{s} is obtained from (9).

The spatial reformulation of the system dynamics above is highly nonlinear due to (9), but is nevertheless appealing for two reasons. First, the required knowledge about the temporal evolution of the \mathcal{T} , \mathcal{N} and \mathcal{B} vectors describing the local Frenet-Serret frame at time t become explicitly available in the integration scheme for ξ' . Secondly, geometric constraints such as a given tracking accuracy transform into simple linear or norm bound constraints. We note that time-varying constraints do not benefit from the spatial reformulation in the same way static constraints do. However, we can impose such constraints by augmenting them as states to the state vector and modeling their trajectories in the prediction model [13].

III. NMPC CONTROL APPROACH

The strength of the system reformulation is clear for a path following task of a robotic manipulator. It allows a geometric optimal control problem formulation for which the shape of the path is described by known parameters to the optimization problem. To illustrate this, a control task is formulated to trade off path following accuracy against tracking a prescribed velocity reference; where path following accuracy is defined as the shortest distance of the end-effector to the path. NMPC is an attractive approach to realize this nonlinear controller task: because it can deal with the nonlinear problem, handle constraints, benefit from existing theoretic results and available efficient solution strategies.

A. Optimal Control Formulation

The quadratic cost function that is considered in this paper is

$$J(\xi(\cdot), \dot{\tau}(\cdot), \cdot) = \int_{\theta=0}^{\theta_f} \varphi(\xi(\theta), \theta)^\top Q \varphi(\xi(\theta), \theta) + \dot{\tau}(\theta)^\top R \dot{\tau}(\theta) d\theta + \varphi(\xi(\theta_f), \theta_f)^\top P \varphi(\xi(\theta_f), \theta_f), \quad (13)$$

where $P, Q \in \mathbb{R}_{\geq 0}^{10 \times 10}$ and $R \in \mathbb{R}_{> 0}^{n \times n}$ denote weighting matrices on the state-dependent vector map $\varphi: \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^{10}$ and the control input $\dot{\tau}$ respectively.

$$\varphi(\xi, \theta) = [r(\xi)^\top, \dot{r}(\xi, \theta)^\top, \phi(\xi)^\top, \dot{s}(\xi, \theta) - \dot{s}_{\text{ref}}(\theta)]^\top, \quad (14)$$

where $\dot{s}_{\text{ref}}(\theta)$ denotes the, possibly pre-optimized [3], velocity profile as a function of the position on the path. In addition to r and the path following velocity error $\dot{s} - \dot{s}_{\text{ref}}$, the orientation ϕ is penalized to prevent non-uniqueness of the solution, while penalizing \dot{r} dampens the error dynamics. The prediction horizon is parameterized with the dimensionless parameter $\theta \in [0, \theta_f]$, due to a rescaling of the system dynamics introduced below. The NMPC control law is defined by the solution of the following minimization problem

$$\underset{\substack{\xi(\cdot) \in \mathbb{R}^m, \\ \dot{\tau}(\cdot) \in \mathbb{R}^n}}{\text{minimize}} \quad J(\xi(\cdot), \tau(\cdot), \cdot) \quad (15a)$$

$$\text{subject to} \quad \frac{d\xi(\theta)}{d\theta} = f_1(\xi(\theta), \dot{\tau}(\theta), \ddot{q}_a(\theta), \theta) \dot{s}_0 h \quad (15b)$$

$$0 = f_2(\xi(\theta), \ddot{q}_a(\theta)) \quad (15c)$$

$$\xi(0) = \hat{\xi}_0 \quad (15d)$$

$$\underline{q}(\theta) \leq q(\theta) \leq \overline{q}(\theta) \quad (15e)$$

$$\underline{\dot{q}}(\theta) \leq \dot{q}(\theta) \leq \overline{\dot{q}}(\theta) \quad (15f)$$

$$\underline{\tau}(\theta) \leq \tau(\theta) \leq \overline{\tau}(\theta) \quad (15g)$$

$$0 < \dot{s}(\xi(\theta), \theta) \quad \forall \theta \in [0, \theta_f], \quad (15h)$$

with the cost function $J(\xi(\cdot), \tau(\cdot), \cdot)$ specified in (13), (15b)-(15c) define the system dynamics. The factor $\dot{s}_0 h$ in (15b) scales the horizon for the measured speed \dot{s}_0 , such that the length of the first spatial prediction interval approximately matches time-based control interval h of the digital controller. The initial condition for the state (15d) is equated to the state measurement $\hat{\xi}_0$. In addition, we have limits on joint angles (15e) to avoid inter-link collisions, and actuator constraints on the joint velocity (15f) and the joint torque (15g). Finally, (15h) ensures forward motion with respect to the nominal path. Other constraints that can be included in (15) are those on $r(\theta)$ (for e.g., collision avoidance or tolerance bounds) and terminal state constraints for stability of the closed-loop control scheme.

In the subsequent paragraph the solution strategy of the proposed NMPC formulation is described.

B. Numerical Solution Strategy

The continuous OCP is transcribed into a nonlinear program (NLP) to obtain a finite-dimensional optimization

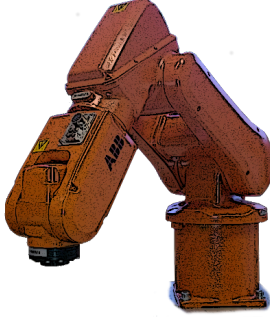


Fig. 2. Sketch of the ABB IRB120 industrial robot.

problem. System dynamics continuity is enforced using a multiple shooting integration strategy [17], path constraints are imposed at each discretization point whereas the cost function is approximated by a finite sum of the Lagrange terms at the multiple shooting nodes. The controls are parameterized as piecewise constant signals.

The concern of real-time implementability of the solution strategy is reflected by evaluating fast optimization methods. To this end the Code Generation tool in the ACADO Toolkit [18] is used to implement the *real-time iteration (RTI) scheme* [19]. The RTI scheme has shown in previous results to compute suboptimal NMPC controls sufficiently fast for real-time implementation. As a crucial component of the SQP-type algorithm the active-set solver qpOASES [20] is used to solve each intermediate quadratic program. ACADO implements efficient differential algebraic equation (DAE) solvers to perform the multiple shooting integration task. This work employs a 4th order implicit Gauss-Legendre method with a fixed number of Newton iterations [21]. The RTI scheme does (by design) not employ any form of globalization; hence stability of the solution scheme is a concern for nonlinear problems such as the path following task presented in this paper. The required expressions for the 6-DOF robot model were generated with the Python-based toolbox SymPyBotics [22]. The Coulomb friction term in (2) is approximated with the continuous differentiable sigmoid function

$$F_s \text{sgn}(\dot{q}) \approx F_s \left(\frac{2}{1 + \exp(-\dot{q}/\varepsilon)} - 1 \right), \quad (16)$$

where the step-like behavior of the sign function is approached for $\varepsilon \rightarrow 0$.

Let us now evaluate the proposed control approach in an illustrative path following task.

IV. SIMULATION RESULTS

One, not necessarily industrial, application of a path-following control task for a robotic manipulator is writing text. To show the efficacy of the NMPC control approach, we present numerical closed-loop simulation results of a robotic manipulator writing the word *OPTEC*. A model of the ABB IRB120 6-DOF industrial manipulator (Fig. 2) [23] is central in the simulations, an experimental setup of this robot is

TABLE I
DENAVIT-HARTENBERG PARAMETERS FOR ABB IRB120.

	d [m]	θ [m]	a [m]	α [rad]
Link 1	0.29	0	0	$-\pi/2$
Link 2	0	$-\pi/2$	0.27	0
Link 3	0	0	0.07	$-\pi/2$
Link 4	0.302	0	0	$\pi/2$
Link 5	0	0	0	$-\pi/2$
Link 6	0.072	π	0	0

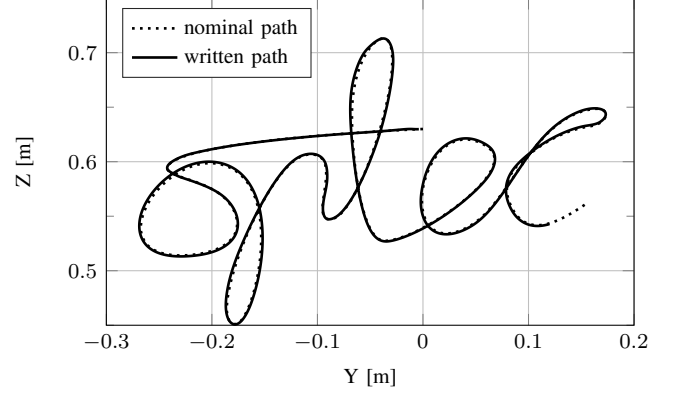


Fig. 3. The written text (*OPTEC*) in scenario 1 projected onto the YZ-plane.

available for future experiments. Dynamic parameters of the robot are provided by ABB Robotics, the Denavit-Hartenberg parameters describing the kinematics are listed in Table I. The control interface with the experimental setup runs at 250 Hz, where the current aim is to update the torque commands every 3 intervals (each 12 ms). To circumvent singularity of the spatial transformation (10) at a standstill, the task is initiated and finalized by a PID controller to respectively launch and brake the robot. Two scenarios are simulated, for two different tunings of the NMPC controller; i.e., for different Q, R, P -matrices in the objective (13). One expresses a preference for accurate tracking over speed, the other is more biased to tracking the velocity reference. Both cases divide the horizon of the OCP in 20 discretization intervals. The weighting matrices for *scenario 1* are

$$Q_1 = \text{blkdiag}(800 \cdot I_3, 30 \cdot I_3, 5 \cdot I_3, 30) \quad (17a)$$

$$R_1 = 10^{-5} \cdot I_3 \quad (17b)$$

$$P_1 = \text{blkdiag}(8000 \cdot I_3, 300 \cdot I_3, 5 \cdot I_3, 300), \quad (17c)$$

where I_n denotes an $n \times n$ identity matrix. And accordingly, the weighting matrices for *scenario 2* are

$$Q_2 = \text{blkdiag}(100 \cdot I_3, 30 \cdot I_3, 5 \cdot I_3, 100) \quad (18a)$$

$$R_2 = R_1 \quad (18b)$$

$$P_2 = \text{blkdiag}(1000 \cdot I_3, 300 \cdot I_3, 5 \cdot I_3, 1000). \quad (18c)$$

The nominal path and the path following simulation results for *scenario 1* are shown in Fig. 3. We observe that the control scheme is able to compute the appropriate torque input to the manipulator to solve the path following problem.

TABLE II
COMPUTATION TIME OF NMPC CONTROL LAW IN SIMULATION. THESE
PRESENT RESULTS GENERATED ON A INTEL CORE I7-4700MQ
NOTEBOOK.

	Average [ms]	Worst case [ms]
Preparation phase	28.09	35.94
Feedback phase	0.781	1.253
Total	28.87	37.19

Penalty on the orientation ϕ of the robot end-effector ensures that the pen remains in the appropriate orientation, joint angles adapt to corners in the nominal path as soon as they appear in the prediction horizon. The total time needed to complete the task is 14.6 s for *scenario 1* and 14.1 s is needed *scenario 2*; the second is shorter in accordance to our expectations, we penalize the velocity tracking error more in *scenario 2*. Fig. 4 compares the simulation results of the two scenarios. The variables \dot{s} , $\|r\|$ and κ show the core of the trade-off of the optimization. Especially at point of high nominal path curvature, the robot slows down considerably in order to maintain close distance to the path. We notice that the velocity is lower for *scenario 1*, which also remains closer to the nominal path. This illustrates one main strength of the path following control approach.

The RTI solution scheme is known for enabling very short computation times for the optimization of the NLP. This is confirmed by the results in Table II. We recognize that solution times need to be further reduced in order to allow real-time implementation. We observe that most time is spent in the so-called *preparation phase* of the solution scheme [19]. This phase linearizes the NLP, of which a major part is the multiple-shooting integration of the DAEs, and condenses the QP in the latest solution iterate. The computation time consumed by qpOASES to solve each intermediate QP in the *feedback phase* is generally shorter than 1 ms, causing this step to contribute little to the total solution time. The real-time iteration scheme is a suboptimal solution scheme, thus not necessarily providing the solution for a local minimum of the NLP at each control instance. Though, as we observe in Fig. 5, the solution remains close to a local optimum as measured by the *KKT-value*. This measure of optimality is commonly used in SQP-algorithms and computed by default in the ACADO Toolkit, a detailed description of the measure is found in e.g., [24]. The optimality measure fluctuates throughout the simulation. It starts of from a very high value, since we do not allow the algorithm to converge before commencing the experiment. Rapid convergence occurs thereafter, with temporary increases of non-optimality appearing mostly at high curvature conditions on the path which requires the algorithm to recover optimality.

We conclude this paper with an overview of the results and present suggestions for future work.

V. CONCLUSIONS AND OUTLOOK

This paper presents a novel NMPC approach for path following control for robotic manipulators. The approach

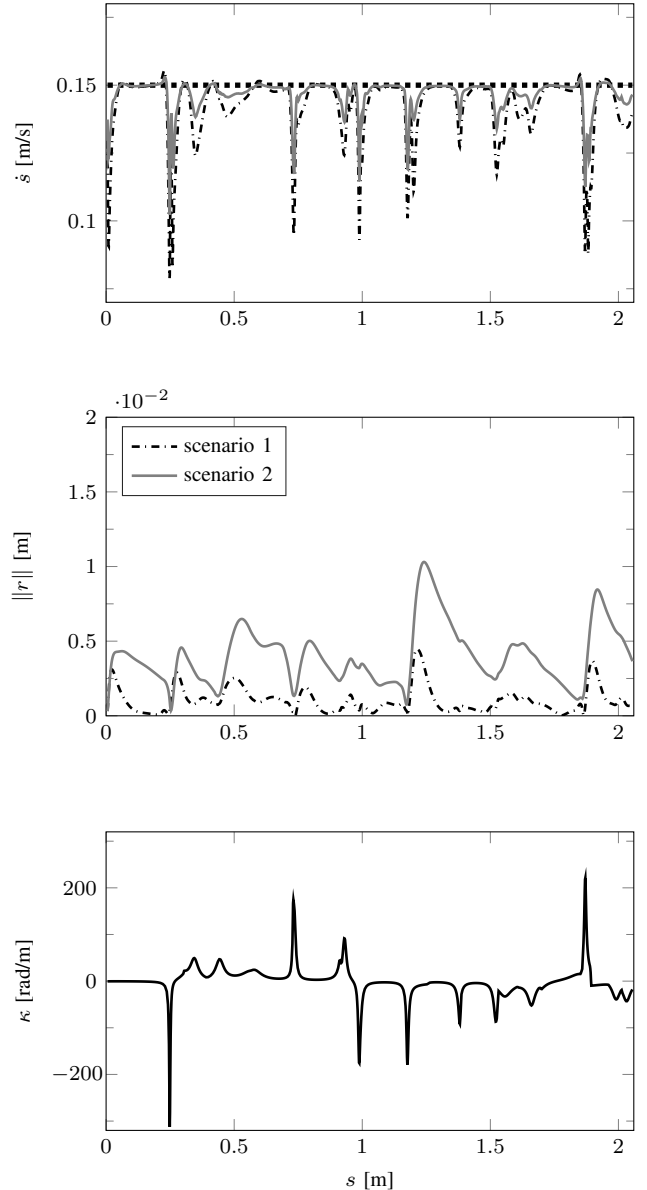


Fig. 4. Comparison of the simulation results of the two scenarios. The top figure shows the evolution of the velocity with respect to the path alongside the reference as a dotted line, the middle figure depicts the distance to the contour and the bottom figure plots the curvature of the path.

is different from existing predictive path following schemes in that it relies on a geometric reformulation of the robot dynamics. The change of variables results in a natural parameterization of the optimal control problem, in which geometric properties of the path appear as known parameters in the spatial prediction horizon. The reformulation also enables an attractive approach to include collision avoidance constraints in the OCP, which will be dealt with in future works. The developed path following NMPC approach is validated numerically for an ABB IRB120 robot executing a writing task. The real-time iteration scheme is implemented using the ACADO Toolkit to generate efficient C-code for our OCP. Simulation results show that the control scheme

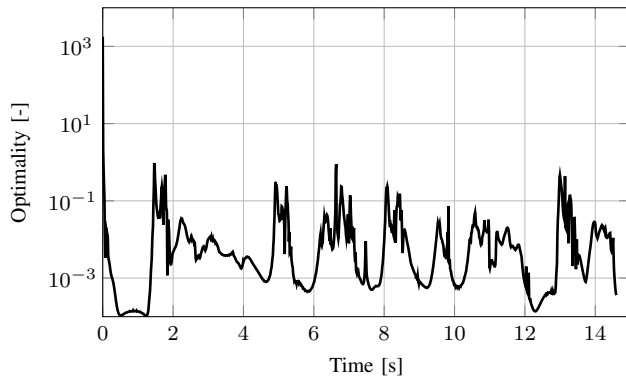


Fig. 5. Evolution of the optimality measure of the control law in time. For the KKT -value converging to zero a KKT -point of the NLP is approached. Fluctuations reflect sub-optimality of the NMPC control law generated in context of the RTI scheme.

can successfully execute a writing task. Different tuning instances of the quadratic cost integral illustrate that the path following controller can effectively trade-off path following accuracy and the velocity at which the writing task is completed. The sub-optimal solution algorithm is able to maintain the decision variables in the near vicinity of the optimum as we observe from the evolution of the KKT -value. Computation times are sufficiently low to enable real-time experiments of the path-following controller for a reduced problem size, such as considering actuation for only three out of the six joints.

Future work consists of establishing conditions for stability and recursive feasibility through computing an appropriate end-state cost and a terminal state set constraint [7]. We aim for a real-time implementation of the presented control scheme with a control interval of 4 ms, it is foreseen that exploiting the possible benefit of parallelization, tuning the integration scheme and modifying the length of the horizon will sufficiently reduce the solution time. In addition, numerical stability of the RTI scheme for the considered problem is of concern and will require further research. With these two issues in mind, experimental verification of the presented work is a short-term goal. Finally, collision avoidance will be formally introduced to the problem formulation to fully benefit from the geometric reformulation at the basis of the proposed control technique. The ability to formulate a collision-free corridor with limited added complexity can aid to exploit the tolerance of a deviation from the nominal path, to reduce the time and energy consumption for robot path following applications.

REFERENCES

- [1] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, Sep. 1985.
- [2] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, Jun. 1985.
- [3] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. Diehl, "Time-optimal path tracking for robots: a convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, pp. 2318–2327, 2009.
- [4] F. Debrouwere, W. V. Loock, G. Pipeleers, M. Diehl, J. Swevers, and J. D. Schutter, "Convex time-optimal robot path following with cartesian acceleration and inertial force and torque constraints," *Proc IMechE Part I: Journal of Systems and Control Engineering*, vol. 227, no. 10, pp. 724–732, Nov. 2013.
- [5] A. P. Aguiar and J. P. Hespanha, "Logic-based switching control for trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," in *Proceedings of the 2004 American Control Conference*, 2004, pp. 3004–3010.
- [6] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotovi, "Performance limitations in reference tracking and path following for nonlinear systems," *Automatica*, vol. 44, no. 3, pp. 598 – 610, 2008.
- [7] T. Faulwasser and R. Findeisen, "Nonlinear model predictive control for constrained output path following," *IEEE Transactions on Automatic Control*, 2015, in press. arXiv:1502.02468.
- [8] T. Faulwasser, B. Kern, and R. Findeisen, "Model predictive path-following for constrained nonlinear systems," in *Joint 48th IEEE conference on Decision and Control and 28th Chinese Control Conference*, 2009.
- [9] D. Lam, C. Manzie, and M. Good, "Model Predictive Contouring Control," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2010.
- [10] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen, "Implementation of nonlinear model predictive path-following control for an industrial robot," 2015, submitted. arXiv:1506.09084.
- [11] Y. Gao, A. Gray, J. V. Frasch, T. Lin, H. E. Tseng, J. Hedrick, and F. Borrelli, "Spatial predictive control for agile semi-autonomous ground vehicles," in *Proceedings of the 11th International Symposium on Advanced Vehicle Control*, 2012.
- [12] J. V. Frasch, A. J. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles," in *Proceedings of the European Control Conference (ECC)*, 2013, pp. 4136–4141.
- [13] N. van Duijkeren, T. Keviczky, P. Nilsson, and L. Laine, "Real-time nmpe for semi-automated highway driving of long heavy vehicle combinations," in *Proceedings of the 5th IFAC Conference on Nonlinear Model Predictive Control 2015 (NMPC'15)*, 2015.
- [14] R. Verschuere, S. D. Bruyne, M. Zanon, J. V. Frasch, and M. Diehl, "Towards time-optimal race car driving using nonlinear MPC in real-time," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2014, pp. 2505–2510.
- [15] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley, 2006.
- [16] H. Guggenheimer, *Differential Geometry*. Dover, 1977.
- [17] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proceedings of the IFAC World Congress*. Pergamon Press, 1984, pp. 242–247.
- [18] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.
- [19] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.
- [20] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.
- [21] R. Quirynen, M. Vukob, and M. Diehl, "Auto generation of implicit integrators for embedded NMPC with microsecond sampling times," in *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*, M. Lazar and F. Allgöwer, Eds., 2012, pp. 175–180.
- [22] C. D. Sousa, "Sympybotics v1.0," Aug. 2014.
- [23] ABB, "IRB120 technical data," 2015, [Online; accessed 22 September 2015].
- [24] D. Leineweber, *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*, ser. Fortschritt-Berichte VDI Reihe 3, Verfahrenstechnik. Düsseldorf: VDI Verlag, 1999, vol. 613.